Amendments to the Claims:

Please replace the claims with the following listing of claims.

1. (Currently Amended) A method for ensuring client access to unpaired messages

from a <u>database management system (DBMS)</u> server, comprising:

the <u>DBMS</u> server ~~distinguishing, by analyzing~~ <u>determining</u> that a <u>transaction</u>

response message <u>is</u>~~, at least one~~ <u>an</u> unpaired message ~~from a paired~~

~~message~~ in response to a communication disruption between ~~the~~ <u>a</u> client

and the <u>DBMS</u> server, the <u>DBMS</u> server storing the ~~at least one~~ unpaired

message in an unpaired message queue, the ~~at least one~~ unpaired message

~~comprising a communication response for~~ <u>corresponding to</u> a specific

client<u>, the transaction response message</u> associated with a transaction

<u>request</u>;

creating the unpaired message queue in a <u>DBMS</u> server <u>separate from a paired</u>

<u>message queue in response to determining a transaction response message</u>

<u>is an unpaired message</u>, the unpaired message queue configured to store a

plurality of unpaired messages intended for the client<u>, the unpaired</u>

<u>message queue and paired message queue storing messages according to a</u>

<u>First-In-First-Out protocol</u>; and

<u>communicating transaction requests and transaction responses using</u> ~~utilizing~~ ~~a~~<u>an</u>

<u>Open Transaction Manager Access (OTMA)</u> protocol which allows the

client to request at least one unpaired message stored in the unpaired

message queue.

2. (Currently Amended) The method of claim 1 further comprising the <u>DBMS</u> server dynamically creating the unpaired message queue in response to the <u>DBMS</u> server detecting at least one unpaired message.

3. (Currently Amended) The method of claim 1, further comprising notifying the <u>DBMS</u> server of a client request to enable dynamic creation of the unpaired message queue.

4. (Currently Amended) The method of claim 3, wherein notifying the <u>DBMS</u> server occurs during establishment of communications between the client and the <u>DBMS</u> server.

5. (Currently Amended) The method of claim 1, further comprising the <u>DBMS</u> server notifying the client when the unpaired message queue contains an unpaired message.

6. (Currently Amended) The method of claim 1, further comprising:

    generating a request message to be sent from the client to the <u>DBMS</u> server; and

    storing an indicator in the request message to enable the client to distinguish between unpaired messages.

7. (Previously Presented) The method of claim 1, wherein utilizing the protocol further comprises allowing the client to request automatic transmission of unpaired messages stored in the unpaired message queue.

8. (Currently Amended) A computer readable medium having stored thereon computer executable instructions for performing a method for ensuring client access to unpaired messages from a <u>database management system (DBMS)</u> server, the method comprising:

the <u>DBMS</u> server ~~distinguishing, by analyzing~~ <u>determining that</u> a <u>transaction</u>

response message <u>is</u>~~, at least one~~ <u>an</u> unpaired message ~~from a paired message~~ in response

to a communication disruption between ~~the~~ <u>a</u> client and the <u>DBMS</u> server, the <u>DBMS</u>

server storing the ~~at least one~~ unpaired message in an unpaired message queue, the ~~at least~~

~~one~~ unpaired message ~~comprising a communication response for~~ <u>corresponding to</u> a

specific client<u>, the transaction response message associated with a transaction request</u>;

creating the unpaired message queue in ~~a~~ <u>the DBMS</u> server<u> separate from a paired</u>

<u>message queue in response to determining a transaction response message is an unpaired</u>

<u>message</u>, the unpaired message queue configured to store a plurality of unpaired messages

intended for a client<u>, the unpaired message queue and paired message queue storing</u>

<u>messages according to a First-In-First-Out protocol</u>; <u>and</u>

<u>communicating transaction requests and transaction responses using</u> ~~utilizing a~~<u>an</u>

<u>Open Transaction Manager Access (OTMA)</u> protocol which allows the client to request

at least one unpaired message stored in the unpaired message queue.

9. (Currently Amended) The computer readable medium of claim 8, wherein the

method further comprising the <u>DBMS</u> server dynamically creating the unpaired

message queue in response to the <u>DBMS</u> server detecting at least one unpaired

message.

10. (Currently Amended) The computer readable medium of claim 8, wherein the

method further comprises notifying the <u>DBMS</u> server of a client request to enable

dynamic creation of the unpaired message queue.

11. (Currently Amended) The computer readable medium of claim 10, wherein notifying the <u>DBMS</u> server occurs during establishment of communications between the client and the <u>DBMS</u> sever.

12. (Currently Amended) The computer readable medium claim 8, wherein the method further comprises the <u>DBMS</u> server notifying the client when the unpaired message queue contains an unpaired message.

13. (Currently Amended) The computer readable medium of claim 8, wherein the method further comprises:

generating a request message to be sent from the client to the <u>DBMS</u> server;

storing an indicator in the request message to enable the client to distinguish between unpaired messages.

14. (Previously Presented) The computer readable medium of claim 8, wherein utilizing the protocol further comprises allowing the client to request automatic transmission of unpaired messages stored in the unpaired message queue.

15. (Currently Amended) A system for ensuring client access to unpaired messages from a <u>database management system (DBMS)</u> server comprising:

a request module configured to receive a client request;

a response generator which receives the client request from the request module and generates an appropriate <u>transaction</u> response message<u> generated based on the client request, the transaction response message associated with a transaction request</u>;

an unpaired message module ~~which distinguishes a paired message from~~ <u>that determines that the transaction response message is</u> an unpaired message ~~based on an analysis of the response message~~ in response to a communication disruption between ~~the~~

a client and the DBMS server and to store paired messages in a paired response data structure and unpaired messages in an unpaired response data structure in response to determining a transaction response message is an unpaired message, the at least one unpaired message comprising a communication response for a specific client, the unpaired response data structure and paired response data structure storing messages according to a First-In-First-Out protocol; and

a response module which communicates paired messages and unpaired messages to a client configure with simple communication logic that conforms to aan Open Transaction Manager Access (OTMA) protocol such that the client constitutes a thin client.

16. (Original) The system of claim 15, wherein the unpaired message module is further configured to dynamically create the unpaired response data structure in response to a first unpaired response message.

17. (Original) The system of claim 15, wherein the response module is configured to automatically send all unpaired messages stored in the unpaired response data structure.

18. (Original) The system of claim 15, wherein the response module is configured to send all unpaired messages stored in the unpaired response data structure in response to a request from the client.

19. (Currently Amended) The system of claim 15, wherein the system is activated upon the DBMS server receiving an activation request from the client.

20. (Original) The system of claim 15, wherein the response module notifies the client when the unpaired response data structure contains at least one unpaired message.